Amendments to the Claims

The listing of claims will replace all prior versions, and listings of claims in the application.

1. (*previously presented*) A method for reducing latency in a sequential record browser, comprising the steps of:

defining a sequential list of records, the records having a retrieval latency;

selecting a record from the list for review;

downloading the selected record, and records ordered sequentially thereafter until interrupted, downloaded records being available for browsing absent the retrieval delay;

interrupting the downloading by selecting a non-sequential record from the list; and

downloading the non-sequential record and records sequentially thereafter until interrupted.

2. (*currently amended*) A method for prequeuing of files predicted to be desired by a user, comprising:

defining a restrictive criteria to select a list of files;

automatically transferring files on the list into a local cache, in anticipation of a user selection thereof, files already transferred to the local cache having a shorter delay for review than those which have not been previously transferred to the local cache,

an order of file transfer being responsive to a prediction of user review requirements, the prediction being responsive to any change in a user deviation from the predicted order; and

receiving a starting point within the list of files, for file review, from the user, such that predicted latencies for sequential file review from any given starting point are ~~optimized~~ reduced.

3. (*previously presented*) The method according to claim 1, wherein said method is executed by a browser application.

4. (*previously presented*) The method according to claim 1, wherein said method is executed by a browser plug-in or extension.

5. (*previously presented*) The method according to claim 1, further comprising the step of cost accounting for downloading of each record.

6. (*previously presented*) The method according to claim 1, further comprising the steps of:

communicating through a network to a server hosting the records; and

presenting a list of records to a user prior to receiving a selection of a record from the user.

7. (*previously presented*) The method according to claim 1, further comprising the steps of:

accounting for a downloaded record; and

limiting said downloading based on a predetermined parameter.

8. (*previously presented*) The method according to claim 2, wherein predicted latencies are minimized.

9. (*currently amended*) The method according to claim 2, wherein the transferring of files is ~~optimized based on both~~ executed to reduce predicted latencies [and] or increase a throughput of the connection between a source of the files being transferred and the local cache.

10. (*currently amended*) The method according to claim 2, wherein the transferring of files is ~~optimized based on both~~ executed to reduce predicted latencies [and] or implement an apparent strategy for review of records by the user.

11. (*currently amended*) The method according to claim 2, wherein the transferring of files is ~~optimized based on both~~ executed to reduce predicted latencies [and] or a cost of the record downloads.

12. (*currently amended*) The method according to claim 2, wherein the transferring of files is ~~optimized based on both~~ executed to reduce predicted latencies [and] or a cost of on-line time.

13. (*currently amended*) The method according to claim 2, wherein the transferring of files is ~~optimized based on both~~ executed to reduce predicted latencies [and] or increase a value of the user's time.

14. (*currently amended*) The method according to claim 2, wherein the transferring of files is ~~optimized based on both~~ executed to reduce predicted latencies [and] or a burden on the server.

15. (*currently amended*) A browser, comprising:

a first interface for defining a restrictive criteria to define a list identifying a set of objects;

logical elements for enabling automatic transferring of an object identified in the list into a cache local to a user, in advance of an actual selection of an object by the user, objects already transferred to the local cache having a lower latency than those which have not been previously transferred to the local cache, an order of object transfer being responsive to a prediction of user requirements, the logical elements being adaptive to a user deviation from the predicted order; and

a second interface for receiving a selection of an object as a starting point within the list of objects, such that predicted latencies for sequential object browsing from any given starting point are ~~optimized~~ reduced.

16. (*previously presented*) The browser according to claim 15, further comprising an accounting system for accounting for downloading of each object.

17. (*previously presented*) The browser according to claim 15, wherein predicted latencies are minimized.

18. (*currently amended*) The browser according to claim 15, wherein the transferring of objects is ~~optimized based on both~~ executed to reduce predicted latencies [and] or increase a throughput of the connection between a source of the objects being transferred and the local cache.

19. (*currently amended*) The browser according to claim 15, wherein the transferring of objects is ~~optimized based on both~~ executed to reduce predicted latencies [and] or a cost associated with the object transfers.

20. (*currently amended*) The browser according to claim 15, wherein the transferring of objects is ~~optimized based on both~~ executed to reduce predicted latencies [and] or increase a value of the user's time.

21. (*previously presented*) A method for transferring files for sequential review, comprising:

accessing a restrictive criteria to select a list of files;

determining an order of file transfer based on a sort criterion;

queuing the files on the list according to the order of file transfer; and

transferring automatically the queued files in a sequential order into a local cache for sequential review at a client.

22. (*previously presented*) A method according to claim 21, wherein said transferring comprises:

receiving a user-specified quantity of files to be transferred; and

transferring the user-specified quantity of files into the local cache.

23. (*previously presented*) The method according to claim 21, further comprising:

receiving a format change during said transferring; and

transferring subsequent files on the list according to the format change.

24. (*previously presented*) The method according to claim 21, further comprising:

receiving a revised sort criterion from the client;

determining a revised order of file transfer based on the revised sort criterion; and

queuing the files on the list according to the revised order of file transfer.

25. (*previously presented*) The method according to claim 21, further comprising:

sending the list to the client; and

receiving, from the client, a first selection of a file from the list, wherein said queuing comprises queuing the first selection as the first file and queuing subsequent files according to the order of file transfer.

26. (*previously presented*) The method according to claim 21, further comprising:

receiving a request to cancel an item from the list; and

removing the item from at least one of the list or the queued files.

27. (*previously presented*) The method according to claim 21, further comprising receiving a non-sequential request for an item on the list, wherein said queuing comprises queuing the requested item as the first file and queuing subsequent files according to the order of file transfer.

28. (*previously presented*) The method according to claim 21, further comprising:

receiving a user-specified image resolution format or a user-specified file format for the files; and

processing the files to comply with the image resolution format or the file format prior to executing said transferring.

29. (*previously presented*) The method according to claim 21, further comprising receiving a user-defined parameter to establish a size of a queue for holding the queued files.

30. (*previously presented*) The method according to claim 29, further comprising transferring, into the local cache, all files held in the queue at once.